UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

---

***Ex parte*** Shawn Joseph Baranczyk and Brian Robert Muras

---

Appeal No. _____
Application No. 10/671,343

---

# APPEAL BRIEF

---

Attorney Docket No. ROC920030213US1 PATENT
Confirmation No. 7069

**CERTIFICATE OF ELECTRONIC TRANSMISSION**
I hereby certify that this correspondence for Application No. 10/671,343 is being electronically transmitted to Technology Center 2132 via EFS-WEB, on August 20, 2008.

/Scott A. Stinebruner/ August 20, 2008
Scott A. Stinebruner, Reg. No. 38,323 Date

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | | |
|---|---|---|
| Applicant: | Shawn Baranczyk et al. | Art Unit: 2132 |
| Application No.: | 10/671,343 | Examiner: Farid Homavounmehr |
| Filed: | September 25, 2003 | |
| For: | ENCRYPTION OF QUERY EXECUTION DETAILS IN A DATABASE MANAGEMENT SYSTEM | |

Mail Stop Appeal Brief - Patents
Commissioner for Patent
P.O. Box 1450
Alexandria, VA 22213-1450

# APPEAL BRIEF

## I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation, of Armonk, New York.

## II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

## III. STATUS OF CLAIMS

Claims 1-11, 13-24 and 26-29 are pending, stand rejected, and are now on appeal. Claims 12, 25 and 30 are canceled.

## IV. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the rejection mailed May 12, 2008 (which was the second rejection of the claims as they currently stand).

## V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant's invention is generally directed to a database monitor that logs execution of a query in a database management system with one or more execution details encrypted or otherwise scrambled to restrict access to those execution details by unauthorized parties. (Application, page 4, lines 2-5). As a result, execution details such as the actual query statements, and values passed to parameter markers and/or host variables associated with the queries executed on a database management system may be kept confidential and accessible only to authorized parties having the ability to decrypt the execution details. (Application, page 4, lines 5-9).

A database monitor is used in many database management systems (DBMS's) to generate an execution log that records each query, as well as a number of execution details about the execution of that query, e.g., the type of access plan used to execute the query, the time and amount of resources used to execute the query, what indexes and other available resources were used during execution of the query, the number of records returned as a result of the query, etc. (Application, page 2, lines 9-15). Additional execution details that may also be logged include the actual query statement, as well as host variables and parameter markers used or incorporated in the query statement. A host variable is typically a variable defined in a host language, or that is generated by an SQL precompiler, and represents data to be inserted into the query during execution. A parameter marker, which is similar in function to a host variable, is typically used in dynamic SQL statements to represent positions in an SQL statement where data is to be inserted by an application. (Application, page 3, lines 1-7).

A database monitor execution log can be used to identify sub-optimal system settings to permit those settings to be adjusted to improve performance, as well as to identify the potential for other performance gains, e.g., to identify particular tables that are accessed with enough frequency to warrant the creation of additional indexes. (Application, page 2, lines 18-22). A

database monitor execution log can also be used to optimize particular queries, e.g., to identify how to alter an original query to improve its execution on the database management system. (Application, page 2, lines 23-27).

Conventional database monitors have been found to suffer from the drawback that the contents of an execution log are stored in an unencrypted format. Some execution details, however, may include confidential and/or proprietary information that a party may not wish to be accessible to others. For example, some execution details such as the values passed to parameter markers or host variables during execution of a query may include confidential information such as employee social security numbers, salary information, etc. (Application, page 3, lines 10-15). Also, the query statements themselves may be confidential to an application developer, as many application developers expend substantial resources to develop complex SQL statements that perform complex operations in a highly tuned manner, and the ability of customers or competitors to view those statements when stored in an execution log could result in the loss of that valuable intellectual asset. (Application, page 3, lines 16-22).

Embodiments consistent with the invention address this problem associated with conventional database monitors by encrypting one or more details in an execution log to restrict access to such execution details by unauthorized parties. (Application, page 4, lines 2-5). The execution of a query is logged in a database management system by generating an encrypted representation of an execution detail for a query executed by the database management system (Application, page 11, lines 23-25; Fig. 3, block 74) and logging the execution detail for the query in an execution log for the database management system by storing the encrypted representation thereof in the execution log (Application, page 11, lines 25-27; Fig. 3, block 70). In addition, the execution detail is decrypted in association with displaying the execution log (Application, page 12, lines 14-17; Fig. 4, blocks 86, 88).

For the convenience of the Board, each of the independent claims has been reproduced below and annotated with references to the specification and drawings to satisfy the requirement to concisely explain the claimed subject matter:

1. A method of executing a query (Application, page 11, lines 7-8) in a database management system (Application, page 8, lines 20-21; Fig. 1, block 28), the method comprising:

receiving an SQL statement from an application program coupled to the database management system (Application, page 11, lines 7-9; Fig. 3, block 60);

executing the SQL statement (Application, page 11, lines 12-14; Fig. 3, block 62);

encrypting the SQL statement to generate an encrypted representation of the SQL statement (Application, page 6, lines 5-7, page 11, lines 23-25; Fig. 3, block 74);

logging execution of the SQL statement in a database monitor by storing the encrypted representation of the SQL statement in an execution log managed by the database monitor (Application, page 11, lines 25-27; Fig. 3, block 70);

displaying the execution log, including retrieving the encrypted representation of the SQL statement from the execution log (Application, page 12, lines 3-5; Fig. 4, block 82), decrypting the encrypted representation of the SQL statement to generate an unencrypted representation of the SQL statement (Application, page 12, lines 14-17; Fig. 4, block 88), and displaying the unencrypted representation of the SQL statement (Application, page 12, lines 12-17; Fig. 4, block 86).

3. A method of logging query execution (Application, page 6, lines 2-5) in a database management system (Application, page 8, lines 20-21; Fig. 1, block 28), the method comprising:

generating an encrypted representation of an execution detail for a query executed by the database management system (Application, page 11, lines 23-25; Fig. 3, block 74);

logging the execution detail for the query in an execution log for the database management system by storing the encrypted representation thereof in the execution log (Application, page 11, lines 25-27; Fig. 3, block 70); and

decrypting the execution detail in association with displaying the execution log (Application, page 12, lines 14-17; Fig. 4, blocks 86, 88).

16. An apparatus (Application, page 7, lines 4-6; Fig. 1, block 10), comprising:

at least one processor (Application, page 7, lines 14-15; Fig. 1, block 12);

a memory (Application, page 7, lines 15-16; Fig. 1, block 14) within which is stored an execution log (Application, page 10, lines 19-22; Fig. 2, block 54); and

program code configured to be executed by the at least one processor to log query execution (Application, page 6, lines 2-5) in a database management system (Application, page 8, lines 20-21; Fig. 1, block 28) by generating an encrypted representation of an execution detail for a query executed by the database management system (Application, page 11, lines 23-25; Fig. 3, block 74), logging the execution detail for the query in the execution log by storing the encrypted representation thereof in the execution log (Application, page 11, lines 25-27; Fig. 3, block 70), and decrypting the execution detail in association with displaying the execution log (Application, page 12, lines 14-17; Fig. 4, blocks 86, 88).

29. A program product (Application, page 9, lines 3-20), comprising:

program code (Application, page 9, lines 3-20) configured upon execution to log query execution (Application, page 6, lines 2-5) in a database management system (Application, page 8, lines 20-21; Fig. 1, block 28) by generating an encrypted representation of an execution detail for a query executed by the database management system (Application, page 11, lines 23-25; Fig. 3, block 74), and logging the execution detail for the query in an execution log for the database management system by storing the encrypted representation thereof in the execution log; and

a recordable computer readable medium (Application, page 9, lines 3-20) storing the program code.

Other support for the claimed subject matter may generally be found in Figs. 3 and 4 and the accompanying text at pages 11-12 of the Application as filed. In addition, it should be noted that, as none of the claims recite any means plus function or step plus function elements, no identification of such elements is required pursuant to 37 CFR §41.37(c)(1)(v). Furthermore, there is no requirement in 37 CFR §41.37(c)(1)(v) to provide support for the subject matter in the separately argued dependent claims, as none of these claims recite means plus function or step plus function elements, and so no discussion of any of these claims is provided.

## VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claim 29 is rejected under 35 U.S.C. §112, second paragraph as being indefinite.

B. Claim 29 is rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter.

C. Claims 1-3, 6-10, 14-16, 19-23 and 27-29 are rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,713,018 to Chan (hereinafter "Chan") in view of U.S. Patent No. 6,289,379 to Urano et al. (hereinafter "Urano").

D. Claims 4-5, 11, 13, 17, 18, 24 and 26 are rejected under 35 U.S.C. §103(a) as being unpatentable over Chan and Urano, and further in view of Official Notice.

## VII. ARGUMENT

Applicant respectfully submits that the Examiner's rejections of claims 1-11, 13-24 and 26-29 are not supported on the record, and should be reversed. Applicant will hereinafter address the Examiner's rejections in the general order presented in the Final Office Action. Within the discussion of each rejection, the various claims that are the subject of the Examiner's rejections will further be addressed in order, starting with the independent claims, and then addressing various dependent claims reciting additional subject matter that is distinguishable from the prior art of record. In some cases, specific discussions of particular claims are not made in the interests of streamlining the appeal. The omission of a discussion with respect to any

particular claim, however, should not be interpreted as an acquiescence as to the merits of the Examiner's rejection of the claim, particularly with respect to claims reciting features that are addressed in connection with the rejections applied to other claims pending in the appeal.

A. Claim 29 is not indefinite.

The Examiner rejects claim 29 as being indefinite under 35 U.S.C. §112, second paragraph, arguing that the specification does not define the term "a recordable computer readable medium." The Examiner acknowledges that the specification defines a "recordable type media," but is apparently of the belief that this definition does not adequately support the term used in the claim.

Applicant is aware of no requirement that every term in a claim be explicitly defined in the specification, so the Examiner's assertion that the claimed term is not explicitly defined is not dispositive of the issue. Applicant submits that the specification as a whole, taken along with the knowledge of one of ordinary skill in the art, provides adequate support for the term. First, as the Examiner acknowledges, the term "recordable type media" is explicitly defined, and it is defined in such a manner that limits the term to only physical, tangible media within the context of statutory subject matter (as discussed in greater detail below). "Media" and "medium" are interchangeable terms, and the term "type" has little bearing on the meaning of the term, so the term "recordable type media" is for all intents and purposes equivalent to the term "recordable medium." The term objected to by the Examiner also qualifies the medium as being "computer readable;" however, this qualifier is self-explanatory, and furthermore is so commonly required by the Office in software-related claims to render a claim statutory, that Applicant submits that one of ordinary skill in the art would have no problem whatsoever ascertaining the meaning of the term "recordable computer readable medium." The term is not indefinite, so reversal of the Examiner's rejection is respectfully requested.[1]

[1] Applicant does note, however, that Applicant would be amendable to amending claim 29 to recite a "recordable type media" in order to overcome the §112 and §101 issues, should the Board decide that the claim is otherwise allowable over the art cited by the Examiner. Should the Examiner wish to make an Examiner's Amendment to render these issues moot, and simplify the issues under appeal, the Examiner may also contact the undersigned.

B. Claim 29 is directed to statutory subject matter.

Claim 29 recites, in part "[a] program product, comprising . . . program code configured upon execution to . . . and; a recordable computer readable medium storing the program code." Applicant specifically defines a "recordable medium" in the specification, at page 9, lines 16-20. This definition makes it clear that a "recordable computer readable medium" includes only tangible or physical type media, and specifically excludes signal-type media such as digital or analog communication links. The relevant passage in the specification, which defines both a "computer readable medium" and a "recordable medium," is reproduced below, with additional paragraph breaks inserted to better illustrate that a "computer readable medium" includes recordable type media and transmission type media, but that recordable type media do not include transmission type media:

> *Examples of computer readable signal bearing media include but are not limited to*
>
> *recordable type media such as volatile and non-volatile memory devices, floppy and other removable disks, hard disk drives, magnetic tape, optical disks (e.g., CD-ROMs, DVDs, etc.), among others, and*
>
> *transmission type media such as digital and analog communication links.* (Application, page 9, lines 16-20).

A medium that is "recordable" as defined in the specification includes only tangible and physical media (volatile and non-volatile memory devices, floppy and other removable disks, hard disk drives, magnetic tape, optical disks ), and by virtue of the separate definition of "transmission type media," specifically excludes the types of media that the Examiner is alleging to be non-statutory.

The Examiner specifically argues in the Office Action that claim 29 is not statutory despite the language in the claim because according to the Examiner the term "recordable computer readable medium" is not defined in the specification, so it is unclear if the term encompasses signals. However, it should be apparent that the operative term being defined in the aforementioned passage is the term "recordable," as the rest of the term includes either a non-limiting word ("type") or the generic term that the term "recordable" is limiting ("media"). The term recited in the claim likewise qualifies the term "medium" by the term "recordable," and by

doing so, similarly limits the claim to the types of media that the Office considers to be statutory. Therefore, Applicant submits that the fact that the claimed term differs to some extent from that defined in the specification, the differences are meaningless within the context of §101.

Applicant therefore respectfully submits that claim 29 is directed to statutory subject matter. Reversal of the Examiner's §101 rejection is therefore respectfully requested.

C. Claims 1-3, 6-10, 14-16, 19-23 and 27-29 are patentable over Chan and Urano.

Applicant respectfully submits that the Examiner's §103(a) rejections of claims 1-3, 6-10, 14-16, 19-23 and 27-29 based upon Chan and Urano are not supported on the record, and should be reversed, given that the Examiner has failed to establish a *prima facie* case of obviousness as to any of these claims. A *prima facie* showing of obviousness requires that the Examiner establish that the differences between a claimed invention and the prior art "are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art." 35 U.S.C. §103(a). Such a showing requires that all claimed features be disclosed or suggested by the prior art. Such a showing also requires objective evidence of the suggestion, teaching or motivation to combine or modify prior art references, as "[c]ombining prior art references without evidence of such a suggestion, teaching or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability -- the essence of hindsight." *In re Dembiczak*, 50 USPQ2d 1614, 1617 (Fed. Cir. 1999).

The rejection of each pending claim will be addressed hereinafter, starting first with the independent claims, and followed by the dependent claims.

Independent Claim 1

Independent claim 1 generally recites a method of executing a query in a database management system. The claimed method comprises receiving an SQL statement from an application program coupled to the database management system; executing the SQL statement; encrypting the SQL statement to generate an encrypted representation of the SQL statement; and logging execution of the SQL statement in a database monitor by storing the encrypted representation of the SQL statement in an execution log managed by the database monitor. The

claim additionally recites displaying the execution log, which includes retrieving the encrypted representation of the SQL statement from the execution log, decrypting the encrypted representation of the SQL statement to generate an unencrypted representation of the SQL statement, and displaying the unencrypted representation of the SQL statement.

In rejecting claim 1, the Examiner has withdrawn his prior rejection based solely on Chan, admitting that Chan does not disclose logging execution of an SQL statement in a database monitor by storing the encrypted representation of the SQL statement in an execution log managed by the database monitor. However, in the new rejection, the Examiner has now added a secondary reference, Urano, arguing that this concept is disclosed in the abstract, col. 1, lines 9-11 and 60-68, col. 4, lines 44-52, and col. 7, lines 35-41 of Urano. The Examiner also continues to support the rejection with col. 1, lines 65-67, col. 2, lines 48-67, col. 3, lines 11-60, col. 4 lines 12-60 and Fig. 4 of Chan.

Applicant submits, however, that the combination of Chan and Urano fails to establish a *prima facie* case of obviousness as to claim 1.

First, as the Examiner does not dispute, Chan does not disclose any database monitor or execution log managed by a database monitor, much less a database monitor or execution log that encrypts any information stored therein. A database monitor is a well known program used in a database management system to log the execution details of the system in an execution log, and is often used to optimize the system and/or the queries processed by the system. As discussed, for example, at page 3 of the Application, a database monitor may store execution details such as the values passed to parameter markers or host variables during execution of a query, as well as the SQL statements that are executed by the database management system.

Urano, however, does not remedy this shortcoming of Chan. Urano is directed to a distributed logging system that monitors abnormal behavior in a computer system, but there is no disclosure or suggestion in the reference of a database monitor, or in particular an execution log that is managed by a database monitor. The reference discloses databases, e.g., as shown in Figs. 1-3 and 10; however, these databases are only discussed in the context of storing log data, or storing analysis rules. There are no particulars in Urano regarding the monitoring of those databases themselves. Put another way, the databases in Urano are part of the logging system, rather than part of the system being logged. Urano in fact neither discloses nor suggests the

monitoring of a database management system, and as such, Applicant submits that the proposed combination falls short of disclosing or suggesting each feature of claim 1, and the rejection of the claim should be reversed.

Second, even if some component in either Chan or Urano were analogized to a database monitor or an execution log managed thereby, neither reference discloses or suggests logging execution of an SQL statement in a database monitor by storing the encrypted representation of the SQL statement in an execution log managed by the database monitor.

As is also discussed, for example, at page 3 of the Application, it has been found that in some instances, some of the execution details logged in an execution log by a database monitor may contain confidential information that a party may not wish to have accessible by others. For example, some execution details such as the values passed to parameter markers or host variables during execution of a query may include confidential information such as employee social security numbers, salary information, etc. In addition, query statements such as SQL statements may be confidential to an application developer, and a developer may prefer to keep those statements from being viewed by customers or competitors. Conventional database monitors, however, typically log all execution details for an executed query execution in an execution log, without regard to the type of execution detail. Consequently, if database monitoring is enabled, the potential exists that confidential information could be logged in a database monitor execution log and accessed by unauthorized parties.

The invention recited in claim 1 addresses this problem in part by encrypting a SQL statement that is being processed by a database management system such that an encrypted representation of that SQL statement is logged in the database monitor. Subsequently, whenever the database monitor is accessed to display all or a portion of the execution log, the SQL statement must be decrypted prior to being displayed. As a result, only parties authorized to decrypt the SQL statement will be able to view an unencrypted representation of the SQL statement when the execution log is displayed. Unauthorized parties, e.g., parties not having the appropriate decryption key, will not be able to view the unencrypted representation of the SQL statement.

Chan lacks any disclosure of encrypting an SQL statement and storing the encrypted SQL statement in an execution log of a database monitor. In fact, neither the term "log" nor the term

"monitor" is even found in the reference. The passages cited by the Examiner merely disclose encrypting SQL statements passed by clients to a database management system as a mechanism for allowing only trusted clients to access certain types of SQL statements (e.g., untrusted clients may only be able to issue read-type SQL statements, but trusted clients may be permitted to issue write-type SQL statements to enable only trusted clients to modify the contents of the database, *see* Chan, col. 1, lines 14-24). The encryption occurs prior to sending a query to a server, and the encrypted query is decrypted prior to executing the query (*see* Chan, Fig. 4, blocks 252 and 258, col. 4, lines 14-40).

Chan does not even disclose any encryption mechanism in a database server. Rather, clients are required to encrypt SQL statements, while a decryption engine in the server is used to decrypt those statements. Even if a conventional monitor is used in the Chan system (though no monitor is actually disclosed), there is no disclosure of any encryption engine within the Chan server that could encrypt SQL statements before they were stored in an encryption log. Of note, the cited passages at col. 3, lines 35-60 and col. 4, lines 50-60 refer to encryption operations performed in clients rather than any server in which a database monitor could be implemented.

Urano does little to address the shortcomings of Chan. Urano discloses the encryption of a log at col. 7, line 36 to col. 8, line 12, for the purpose of preventing alteration of a log. In the Urano process, a log is divided into multiple portions, given a digital signature and encrypted, with each portion sent to a different computer for storage. It is important to note that each computer then decrypts the received log and saves the decrypted version for storage (col. 7, lines 60-63). It is somewhat unclear as to whether an encrypted version of a log is even ever stored in an execution log in Urano, since the log, once sent to another computer, is decrypted prior to storage. Moreover, since Urano states at col. 7, lines 60-63 that an end user computer 1011 typically accesses computers 1005, 1006 and 1007 to read the log (Fig. 10), and the log is decrypted in those computers, Urano is unconcerned with restricting access to a log, or to preserving the confidentiality of the details in the log.

Furthermore, given that Urano is not specifically directed to database monitors or execution logs managed thereby, Urano cannot be relied upon to suggest encrypting and storing database-specific execution details such as SQL statements within an execution log, as required by claim 1. Therefore, neither Chan nor Urano discloses or suggests the concept of encrypting

an SQL statement and storing the encrypted representation thereof in an execution log managed by a database monitor. Applicant submits that the proposed combination therefore falls short of disclosing or suggesting this additional feature of claim 1, and the rejection of the claim should be reversed.

Third, claim 1 also recites the display of the execution log, along with the retrieval, decryption and display of an unencrypted representation of an SQL statement that has been encrypted and stored in the execution log. In rejecting this feature, the Examiner implicitly admits that Chan does not disclose this feature, and instead relies on col. 7, line 60 to col. 8, line 2 of Urano, arguing that the passage discloses decryption and display of logs to an administrator for error or attack detection.

As noted above, however, it is not even clear that Urano stores an encrypted information in a log, since the encryption is performed on one computer, the encrypted information is sent to multiple other computers, and those computers decrypt the information and store the information in decrypted form (col. 7, lines 60-63). It is only after the information is decrypted and stored in one of the computers that a user can access the information (col. 7, lines 64-66).

Irrespective of this fact, however, Urano still does not disclose decrypting an SQL statement stored in a log and displaying that decrypted statement to a user. Applicant is not merely claiming the display of an execution log. Instead, claim 1 is directed to displaying an SQL statement in an unencrypted form, and in connection with the display of an execution log, after than SQL statement has been encrypted and stored in the execution log. As Applicant noted above, in many cases an application developer may have a substantial interest in maintaining the confidentiality of his or her proprietary SQL statements. A developer's application, moreover, may issue queries to a database management system that is not under the control of the application developer, and as such, a system administrator for the database management system (who is typically not the application developer) conventionally can view the developer's SQL statements in unencrypted form in an execution log, which effectively destroys the confidentiality of the proprietary SQL statements. By encrypting the SQL statements in the execution log, and then decrypting such SQL statements in association with displaying the log, access to the SQL statements may be limited to those possessing the proper authorization, thus enabling a developer to maintain the confidentiality of his or her SQL proprietary work product.

It should be noted that neither Chan nor Urano appreciates the problems addressed by the invention of claim 1, and neither proposes any analogous solution thereto. Chan is directed to encrypting SQL statements to enable only authorized clients to issue certain types of statements to a database management system, and the encrypted SQL statements are decrypted prior to execution by the system. Urano is directed to encrypting a log to prevent alteration of the log, and it appears that no access control is provided to the log once it is stored on one of computers 1005, 1006, 1007, given that the log is apparently stored in decrypted form.

While the references need not necessarily have to suggest Applicant's particular problem, the fact that the references are directed to the use of encryption for completely different reasons is highly relevant to the question of whether the references suggest encrypting SQL statements that are stored in the execution log of a database monitor for a database management system and then decrypting those SQL statements in connection with displaying them to a user. From a fundamental standpoint, neither reference is even directed to database monitors or execution logs managed thereby, so Applicant submits that one of ordinary skill in the art would not be motivated to combine the references, or to modify either reference, to incorporate the encryption and decryption of SQL statements in a database monitor execution log, as would be required to establish a *prima facie* case of obviousness for claim 1.

The Examiner's approach in this case, in fact, is replete with hindsight-based analysis. It appears the Examiner has basically taken the position that because monitoring databases is known, and encrypting data stored in a log is known, then any encryption of database-related information in a log must likewise be known. The Examiner's approach, however, effectively reads out specific features recited in claim 1, most notably the encryption and decryption of SQL statements within a database monitor execution log. No objective reason has been proffered as to why one of ordinary skill in the art would be motivated to modify either reference to encrypt SQL statements in a database monitor execution log for a database management system. Absent any such reason, no *prima facie* case of obviousness can be maintained.

Applicant respectfully submits that Chan and Urano do not disclose or suggest either the encryption of SQL statements that are stored in an execution log of a database monitor, or the decryption of such statements in association with displaying an execution log. Moreover, given the unique and unexpected advantage of protecting potentially confidential execution details

from being accessed via a database monitor, Applicant submits that claim 1 is non-obvious over Chan and Urano. The Examiner, furthermore, has provided no credible evidence of the recognition in the art of an objective reason to modify either reference to include Applicant's claimed database monitor execution log in which SQL statements are encrypted, stored, and then decrypted in association with displaying the execution log. Applicant therefore respectfully requests that the Examiner's rejection of claim 1 be reversed, and that claim 1, as well as claim 2 which depends therefrom, be passed to allowance.

Independent Claim 3

Next, with regard to the rejection of independent claim 3, this claim generally recites a method of logging query execution in a database management system, and recites generating an encrypted representation of an execution detail for a query executed by the database management system, logging the execution detail for the query in an execution log for the database management system by storing the encrypted representation thereof in the execution log, and decrypting the execution detail in association with displaying the execution log.

In rejecting claim 3, the Examiner merely references the rejection of claim 1. As noted above in connection with claim 1, however, neither Chan nor Urano discloses or suggests the encryption of SQL statements or other execution details and the storage of same in an encrypted representation in an execution log. Also, neither reference discloses or suggests decrypting an execution detail in association with displaying an execution log. In particular, neither reference even discloses an execution log for a database management system, as required by claim 3.

Applicant therefore respectfully submits that the Examiner has failed to establish a *prima facie* case of obviousness as to claim 3. Reversal of the Examiner's rejection of claim 3, and allowance of that claim and of claims 4-11 and 13-15 which depend therefrom, are therefore respectfully requested.

Independent Claim 16

Next, with respect to independent claim 16, this claim recites, similar to claim 3, the concept of generating an encrypted representation of an execution detail for a query executed by the database management system, logging the execution detail for the query in the execution log

by storing the encrypted representation thereof in the execution log, and decrypting the execution detail in association with displaying the execution log. As discussed above in connection with claims 1 and 3, neither Chan nor Urano discloses or suggests the concept of encrypting execution details, be they SQL statements or otherwise, within an execution log, as an execution log is described in the specification, nor does either reference disclose or suggest decrypting execution details in an execution log in association with displaying the execution log. Accordingly, claim 16 is non-obvious over Chan and Urano, and the rejection should be reversed. Reversal of the Examiner's rejection of claim 16, and allowance of that claim and of claims 17-24 and 26-28 which depend therefrom, are therefore respectfully requested.

Independent Claim 29

Next, with respect to independent claim 29, this claim recites, similar to claim 3, the concept of generating an encrypted representation of an execution detail for a query executed by the database management system and logging the execution detail for the query in the execution log by storing the encrypted representation thereof in the execution log. As discussed above in connection with claims 1 and 3, Chan and Urano do not disclose or suggest the concept of encrypting execution details, be they SQL statements or otherwise, within an execution log, as an execution log is described in the specification. Accordingly, claim 29 is non-obvious over Chan and Urano. Reversal of the Examiner's rejection of claim 29, and allowance of the claim are respectfully requested.

Dependent Claims 2, 8, 9, 21 and 22

Claim 2 additionally recites encrypting at least one value passed to one of a host variable and a parameter marker used by the SQL statement, and that logging execution of the SQL statement includes storing the encrypted value in the execution log. Similarly, claims 8 and 21 recite that the execution detail comprises a value passed to a host variable during execution of the query, while claims 9 and 22 recite that the execution detail comprises a value passed to a parameter marker during execution of the query.

Host variables and parameter markers are described in the specification at page 3, lines 1-7. A host variable is defined as a variable defined in a host language, or that is generated by an

SQL precompiler, and represents data to be inserted into the query during execution. A parameter marker, which is similar in function to a host variable, is defined as being capable of being used in dynamic SQL statements to represent positions in an SQL statement where data is to be inserted by an application.

In rejecting these claims, the Examiner relies on col. 3, lines 12-60 of Chan, arguing that the SQL statement, which is encrypted and sent during execution, corresponds to a value for a host variable. However, it is important to note that each of claims 2, 8, 9, 21 and 22 recites a value that is either "used by the SQL statement," (claim 2) or "passed . . .during execution of the query." The Examiner's use of the SQL statements themselves as corresponding to "values" effectively reads the fact that values are provided in addition to SQL statements or queries, out of these respective claims. Furthermore, as disclosed at col. 3, lines 50-60, the values that are passed as arguments for placeholders in Chan are unencrypted – only the SQL statement string is encrypted. Thus, Applicant submits that even to the extent that Chan discloses encrypting an SQL statement, the encrypted SQL statement does not correspond to the values claimed in claims 2, 8, 9, 21 and 22. Furthermore, given that the SQL statement in Chan is not even stored in encrypted form in an execution log, or decrypted in association with the display of the execution log, as discussed above in connection with claims 1, 3 and 16, Applicant submits that Chan does not read on these claims. Urano, which is not even directed to a database management system, is irrelevant to any of these concepts, so the reference adds nothing to the rejections in this regard. Reversal of the Examiner's rejections of claims 2, 8, 9, 21 and 22 are therefore respectfully requested.

Dependent Claims 6-7 and 19-20

Claims 6-7 and 19-20 are not argued separately.

Dependent Claims 10 and 23

Claims 10 and 23 additionally recite logging a second execution detail for a query in the execution log in an unencrypted representation. The Examiner's rejection, however, merely discusses the fact that Urano allegedly has one embodiment, described before col. 7, line 35, that does not encrypt logged execution records, and one embodiment that does encrypt logged

execution records. By the Examiner's own admission, however, these variations are in separate embodiments, and Urano does not disclose or suggest that both encrypted and unencrypted details can reside in the same execution log, so the Examiner has failed to establish a *prima facie* case of obviousness as to the combination. The Examiner also states that an unencrypted set of records must be created before an encrypted set can be generated; however, even if that is true, the claim is directed to storing/logging execution details in an execution log, so even if a record is unencrypted before it is encrypted, if the unencrypted form is not logged or stored in the log, the creation of that unencrypted record falls short of disclosing claimed feature of logging an unencrypted execution detail in an execution log.

Applicant therefore respectfully submits that no *prima facie* case of obviousness exists with respect to claims 10 and 23, so reversal of the Examiner's rejections of these claims are respectfully requested.

Dependent Claims 14-15 and 27-28

Claims 14-15 and 27-28 are not argued separately.

D. Claims 4-5, 11, 13, 17, 18, 24 and 26 are patentable over Chan, Urano, and Official Notice.

Dependent Claims 4-5 and 17-18

Claims 4 and 17 are not argued separately.

Dependent Claims 11 and 24

Claims 11 and 24 depend respectively from claims 10 and 23, and additionally recite that the second execution detail (which is logged in unencrypted form) includes at least one of an access plan and a performance statistic associated with execution of the query. Thus, more so than any of the other claims at issue, these claims quite clearly address execution details, logged in unencrypted form, which are related to the execution and performance of a database query.

As is well known in the art, an access plan represents the executable representation of a database query, i.e., the optimized "plan of attack" for executing the query. Likewise, a

performance statistic represents how the access plan performed when executing the query. As disclosed at page 6, lines 13-17, an execution details associated with an access plan may include, for example, join orders, and what indexes and other available resources were used, while performance statistics may include, for example, the time and amount of resources used to execute queries and the number of records returned as a result of queries.

In rejecting claims 11 and 24, the Examiner relies principally on the fact that Urano allegedly discloses collecting logs related to execution processes. The Examiner also notes that Applicant has admitted that performance statistics and access plans are well known in the art. While performance statistics and access plans are well known in the art, storing such statistics and access plans in an execution log for a database monitor in an unencrypted form, while other execution details for a database query are stored in the same execution log in encrypted form, is not disclosed or suggested by the prior art of record. The Examiner's reliance on Official Notice ignores the specific language of the claims, and attempts to parse the language of the claims to such an extent that the totality of the claimed invention is lost. The Examiner has not and can not show a teaching in the prior art related to storing performance statistics and/or access plans for a query unencrypted in a database monitor execution log simultaneously with storing other execution details for that query in the log in encrypted form. Applicant therefore respectfully submits that no *prima facie* case of obviousness exists with respect to claims 11 and 24, so reversal of the Examiner's rejections of these claims are respectfully requested.

Dependent Claims 13 and 26

Claims 13 and 26 are not argued separately.

## CONCLUSION

Applicant respectfully requests that the Board reverse the Examiner's rejections of claims 1-11, 13-24 and 26-29, and that the Application be passed to issue. If there are any questions regarding the foregoing, please contact the undersigned at 513/241-2324. If any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

August 20, 2008
Date

/Scott A. Stinebruner/
Scott A. Stinebruner
Reg. No. 38,323
WOOD, HERRON & EVANS, L.L.P.
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
Telephone: (513) 241-2324
Facsimile: (513) 241-6234

# VIII. CLAIMS APPENDIX: CLAIMS ON APPEAL (S/N 10/671,343)

**Listing of Claims:**

1. (Previously Presented) A method of executing a query in a database management system, the method comprising:

receiving an SQL statement from an application program coupled to the database management system;

executing the SQL statement;

encrypting the SQL statement to generate an encrypted representation of the SQL statement;

logging execution of the SQL statement in a database monitor by storing the encrypted representation of the SQL statement in an execution log managed by the database monitor;

displaying the execution log, including retrieving the encrypted representation of the SQL statement from the execution log, decrypting the encrypted representation of the SQL statement to generate an unencrypted representation of the SQL statement, and displaying the unencrypted representation of the SQL statement.

2. (Original) The method of claim 1, further comprising encrypting at least one value passed to one of a host variable and a parameter marker used by the SQL statement, wherein logging execution of the SQL statement further comprises storing the encrypted value in the execution log.

3. (Previously Presented) A method of logging query execution in a database management system, the method comprising:

generating an encrypted representation of an execution detail for a query executed by the database management system;

logging the execution detail for the query in an execution log for the database management system by storing the encrypted representation thereof in the execution log; and

decrypting the execution detail in association with displaying the execution log.

4. (Original) The method of claim 3, further comprising receiving the query in an unencrypted form from an application program in communication with the database management system.

5. (Original) The method of claim 4, wherein generating the encrypted representation is performed after communicating the query to the database management system.

6. (Original) The method of claim 3, wherein generating the encrypted representation is performed prior to communicating the query to the database management system.

7. (Original) The method of claim 3, wherein the execution detail comprises a query statement.

8. (Original) The method of claim 3, wherein the execution detail comprises a value passed to a host variable during execution of the query.

9. (Original) The method of claim 3, wherein the execution detail comprises a value passed to a parameter marker during execution of the query.

10. (Original) The method of claim 3, further comprising logging a second execution detail for the query in the execution log in an unencrypted representation.

11. (Original) The method of claim 10, wherein the second execution detail includes at least one of an access plan and a performance statistic associated with execution of the query.

12. (Canceled).

13. (Previously Presented) The method of claim 3, wherein generating the encrypted representation includes encrypting the execution detail using a public key, and wherein

decrypting the execution detail includes decrypting the execution detail using a private key paired with the public key.

14. (Original) The method of claim 3, further comprising determining if database monitoring is enabled in the database management system, wherein generating the encrypted representation is performed if it is determined that database monitoring is enabled.

15. (Original) The method of claim 3, wherein the query comprises an SQL statement.

16. (Previously Presented) An apparatus, comprising:

at least one processor;

a memory within which is stored an execution log; and

program code configured to be executed by the at least one processor to log query execution in a database management system by generating an encrypted representation of an execution detail for a query executed by the database management system, logging the execution detail for the query in the execution log by storing the encrypted representation thereof in the execution log, and decrypting the execution detail in association with displaying the execution log.

17. (Original) The apparatus of claim 16, wherein the program code is further configured to receive the query in an unencrypted form from an application program in communication with the database management system.

18. (Original) The apparatus of claim 17, wherein the program code is configured to generate the encrypted representation after communicating the query to the database management system.

19. (Original) The apparatus of claim 16, wherein the program code is configured to generate the encrypted representation prior to communicating the query to the database management system.

20. (Original) The apparatus of claim 16, wherein the execution detail comprises a query statement.

21. (Original) The apparatus of claim 16, wherein the execution detail comprises a value passed to a host variable during execution of the query.

22. (Original) The apparatus of claim 16, wherein the execution detail comprises a value passed to a parameter marker during execution of the query.

23. (Original) The apparatus of claim 16, wherein the program code is further configured to log a second execution detail for the query in the execution log in an unencrypted representation.

24. (Original) The apparatus of claim 23, wherein the second execution detail includes at least one of an access plan and a performance statistic associated with execution of the query.

25. (Canceled).

26. (Previously Presented) The apparatus of claim 16, wherein the program code is configured to generate the encrypted representation by encrypting the execution detail using a public key, and wherein the program code is configured to decrypt the execution detail by decrypting the execution detail using a private key paired with the public key.

27. (Original) The apparatus of claim 16, wherein the program code is further configured to determine if database monitoring is enabled in the database management system, and wherein the program code is configured to generate the encrypted representation if it is determined that database monitoring is enabled.

28. (Original) The apparatus of claim 16, wherein the query comprises an SQL statement.

29. (Previously Presented) A program product, comprising:

program code configured upon execution to log query execution in a database management system by generating an encrypted representation of an execution detail for a query executed by the database management system, and logging the execution detail for the query in an execution log for the database management system by storing the encrypted representation thereof in the execution log; and

a recordable computer readable medium storing the program code.

30. (Canceled).

## IX. EVIDENCE APPENDIX

10/671,343

None.

## X. RELATED PROCEEDINGS APPENDIX

10/671,343

None.